Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2009, Article ID 947823, 19 pages
doi:10.1155/2009/947823

Research Article

Analysis of Damped Mass-Spring Systems for Sound Synthesis

Don Morgan and Sanzheng Qiao

Department of Computing and Software, McMaster University, Hamilton, ON, Canada L8S 4L7

Correspondence should be addressed to Don Morgan, mordon700@aol.com

Received 26 July 2008; Revised 1 December 2008; Accepted 20 February 2009

Recommended by Thippur Sreenivas

There are many ways of synthesizing sound on a computer. The method that we consider, called a *mass-spring system*, synthesizes sound by simulating the vibrations of a network of interconnected masses, springs, and dampers. Numerical methods are required to approximate the differential equation of a mass-spring system. The standard numerical method used in implementing mass-spring systems for use in sound synthesis is the *symplectic Euler* method. Implementers and users of mass-spring systems should be aware of the limitations of the numerical methods used; in particular we are interested in the stability and accuracy of the numerical methods used. We present an analysis of the symplectic Euler method that shows the conditions under which the method is stable and the accuracy of the decay rates and frequencies of the sounds produced.

1. Introduction

Physical sound synthesis uses mathematical models based on the physics of sound production to synthesize sound. In other words, physical sound synthesis uses a model to simulate the sound producing object, rather than the sound produced by an object. In this paper we focus on mass-spring systems: networks of masses, springs, and dampers. The mathematical model of mass-spring systems is based on differential equations. To approximate these differential equations on a digital computer, numerical methods are used. An important question to ask when using a numerical method is how well does this method approximate the differential equations used in the system? Numerical methods can become *unstable*. This means that the numerical solution can deviate arbitrarily far from the exact solution. In many cases the error can grow without bound, making the results of the numerical method meaningless.

As well, we want to be able to quantify the accuracy of our approximation. Most musical sounds are composed of a number of different frequencies. The lowest of these frequencies is called the *fundamental frequency* [1]. The fundamental frequency determines the perceived pitch of the sound, and an error in the fundamental frequency will cause the sound to be out of tune. The higher frequency components influence the *timbre*, or tone colour, of the sound [1] and errors in these components give the sound a different timbre than it should. An error in frequency caused by a numerical method is known as *frequency warping*.

The decay rate determines how quickly the amplitude (or volume) of the sound decreases. For example, a note on a piano can be heard for 20 or 30 seconds after it is struck, while on a banjo it becomes imperceptible after only 3 or 4 seconds because the decay rate of a banjo is much larger than that of a piano. Numerical methods may add extraneous damping to vibrating systems that are undamped. This is known as *numerical damping*.

There have been several sound synthesis systems built using mass-spring systems and described in the literature [2–6]. Most of these systems have used a numerical method called the *symplectic Euler* method. The previous literature on mass-spring systems used in sound synthesis describe how these systems work (i.e., the equations used and the finite difference equations used to approximate them), but have not addressed the issues of the stability and accuracy of the numerical methods. This has been an important part of the criticism of mass-spring systems in the physical synthesis literature. Mass-spring system have been criticized as being computationally expensive [7], lacking an analysis of stability [7], and having an unknown accuracy [8].

The symplectic Euler method has been studied by researchers outside the sound synthesis community. The book by Hairer et al. [9] presents a thorough analysis of symplectic numerical methods including the symplectic Euler method. Using different methods—they do not use the z-transform—they arrive at the stability condition for the undamped mass-spring system as

$$h\omega_0 \leq 2, \tag{1}$$

which is the same as our results when the damping is zero. They do not analyze the symplectic Euler method in terms of frequency warping or its effect on damping. The thesis by Beck [10] contains a proof that symplectic Euler method is symplectic, which implies that it has no numerical damping for an undamped mass-spring system, which agrees with our conclusions. Beck's thesis looks at using the symplectic Euler on the Lotka-Volterra (predator-prey) equations. It also does not analyze its frequency warping or its effect on damping. In a previous paper [11] we have given an analysis of the symplectic Euler method when used to simulate *undamped* mass-spring systems. In this paper, we extend that analysis to include *damped* mass-spring systems.

The contributions of this paper are the equations for the stability, the frequency warping, and numerical damping of damped mass-spring systems presented in Section 3. This paper is not presenting a new method of doing sound synthesis or suggesting improvements to existing methods. The question it proposes to answer is, if given a specification of a mass-spring system, that is, the values for the mass, spring stiffness, and viscous damping constants, and the connections between them—will the system be stable, and if so, what sound will it produce? The three main questions addressed in this paper are the following.

(1) Under what conditions are damped mass-spring systems using the symplectic Euler method stable?

(2) What is the accuracy of the frequencies of the sounds produced by damped mass-spring systems using the symplectic Euler method?

(3) What is the accuracy of the decay rates of the sounds produced by damped mass-spring systems using the symplectic Euler method?

Section 2 introduces the mass-spring system and explains why the symplectic Euler method is often used to discretize the differential equations of a mass-spring system. Section 3 presents the analysis of the symplectic Euler method. We begin by using the symplectic Euler method to discretize a mass-spring system containing only one mass. We use the z-transform to find the symplectic Euler method's effect on the frequency and decay rate of the system, and find the conditions for stability of the system. This section contains the main contributions of the paper. We end this section by demonstrating the consistency of our theoretical results with the results of a computer simulation of a mass-spring system. In Section 4 we show how the results in the previous section can be extended to mass-spring systems with more than one mass. Section 5 concludes with a summary of our results.

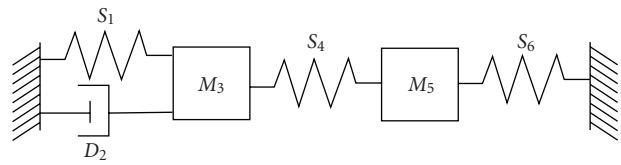

Figure 1: Simple mass-spring system.

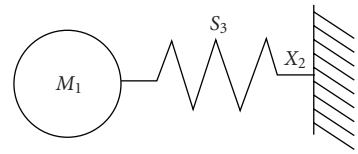

Figure 2: Undamped mass-spring system.

2. The Mass-Spring Model

The *mass-spring* model builds complex musical instruments from simple components: masses, springs, and dampers. Each element is discretized using finite difference methods. The behavior of the system depends solely on the network and the physical equations of each of the components. No other physical equations are used.

Figure 1 shows a simple mass-spring model, where M_3 and M_5 are masses, S_1, S_4, and S_6 are springs and D_2 is a damper.

2.1. Choosing a Numerical Method for Mass-Spring Systems. To simulate the vibrations of a mass-spring system on computer, we need to use a numerical method to discretize the differential equations of the system. There are many numerical methods to choose from. What should we look for when choosing a numerical method for sound synthesis? Humans can hear sounds that have frequencies from 20 to 20 000 Hz. Notes played on typical musical instruments such as a piano, a guitar, a trumpet, and so forth. may last for several seconds. This means that a note may contain thousands or tens of thousands of cycles. The energy of a mass-spring system is the sum of its potential and kinetic energies, which depend on the amplitudes of the vibrations. It is therefore important that a numerical method used in sound synthesis can be able to conserve energy for thousands of cycles. If the numerical method causes the energy to increase over time, the simulation will become unstable. Conversely, if the numerical method causes the energy to decrease, the sound will decay more rapidly than it should. Numerical methods that do not conserve energy have proved to be a problem in fields such as molecular [12] and planetary simulation [13]. There has been an interest in recent years in numerical methods that can accurately simulate the qualitative aspects of physical systems. *Symplectic* numerical algorithms, among other properties, conserve energy over long periods of time [14].

Figure 2 shows an undamped mass-spring system containing one mass and one spring.

If we regard the equilibrium position of the spring to be $x = 0$, the force of the spring, according to Hooke's Law, is $F(t) = -kx(t)$, where k is the spring stiffness coefficient and $x(t)$ is the position of the mass at time t. We can then write the differential equation for the system, using a for the acceleration and m for the mass, as

$$\begin{aligned} ma(t) &= -kx(t) \quad \text{(by Newton's 2nd law)}, \\ \frac{d^2x(t)}{dt^2} &= -\left(\frac{k}{m}\right)x(t). \end{aligned} \tag{2}$$

This is a second-order differential equation. The general solution is [15]

$$x(t) = C_1 \cos\left(\sqrt{\frac{k}{m}}t\right) + C_2 \sin\left(\sqrt{\frac{k}{m}}t\right). \tag{3}$$

Setting initial condition $x(0) = 1$ (the initial position) and $x'(0) = 0$ (the initial velocity) the particular solution is

$$x(t) = \cos(\omega_0 t), \tag{4}$$

where $\omega_0 = \sqrt{k/m}$ is the radial frequency of the system. So the solution of the system is simply a cosine wave of frequency ω_0. Since there is no damping the amplitude of the cosine wave should not change over time (i.e., the system should conserve energy). Next, we examine how well this system is simulated by three first-order numerical methods: the forward Euler, the backward Euler, and the symplectic Euler. In these simulations we set ω_0 at $125 \times 2\pi$ radians per second and the sample rate at 1000 samples per second (i.e., the frequency is 1/8 the sample rate).

The forward Euler method is defined as

$$y_{n+1} = y_n + y'_n h, \tag{5}$$

where y' is the first derivative of y with respect to time and h is the length of each time step. We use the subscript notation to represent numerical approximations (e.g., y_{n+1} denotes the numerical approximation of y at time step $n+1$). The forward Euler method gains energy over time, causing it to be unstable for undamped or lightly damped systems. Figure 3 shows the result of simulating the above mass-spring system using the forward Euler method with ω_0 equal to 1/8 the sampling frequency. We can see that this results in an unstable system.

The backward Euler method is defined as

$$y_{n+1} = y_n + y'_{n+1} h. \tag{6}$$

The backward Euler is called an *implicit* method since it uses the derivative at the new point which has not yet been determined. The backward Euler method loses energy over time. Figure 4 shows the result of simulating the above mass-spring system using the backward Euler method with ω_0 equal to 1/8 the sampling frequency. We can see that this results in cosine wave that is quickly damped.

The symplectic Euler is defined as

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ v_n \end{pmatrix} + h \begin{pmatrix} v_{n+1} \\ a_n \end{pmatrix}, \tag{7}$$

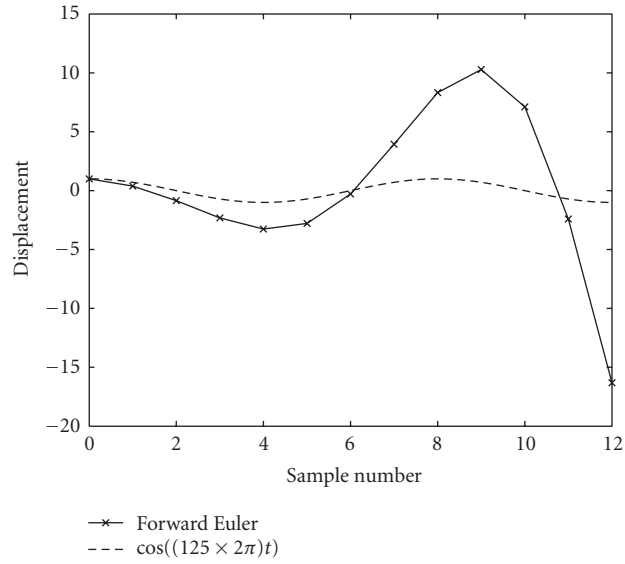

Figure 3: Forward Euler approximation of the undamped mass-spring system ($h = .001$).

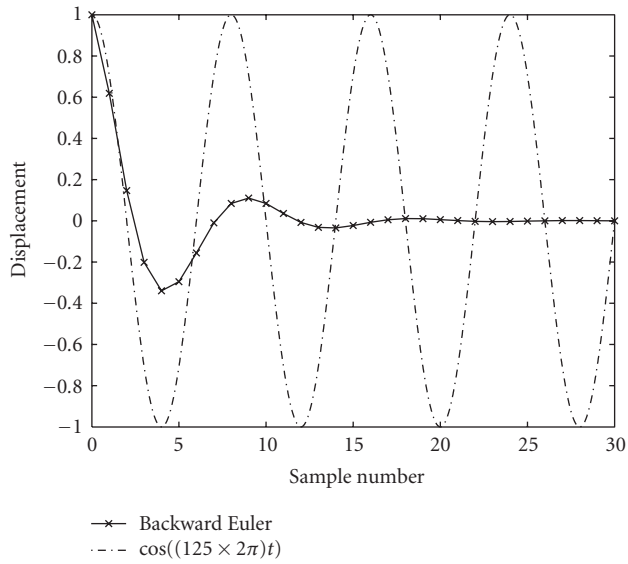

Figure 4: Backward Euler approximation of the undamped mass-spring system ($h = .001$).

where x is the displacement, v is the velocity, and a is the acceleration. We first calculate the new velocity v_{n+1}, since it can be calculated using the known acceleration a_n and the known velocity v_n. We can then use v_{n+1} to calculate the new position, x_{n+1}. This makes the entire system *explicit*. This method is therefore sometimes referred to as the *explicit* version of the symplectic Euler [10], since there is also an implicit version [9, 10]. Since the symplectic Euler method, as its name implies, has been shown to be a symplectic numerical method [9, 10] unlike the forward and backward Euler methods, it should conserve energy. Figure 5 shows the result of simulating the above mass-spring system using the symplectic Euler method with ω_0

equal to 1/8 the sampling frequency. We can see that the results of this simulation are much nearer to the analytic solution than either the forward or backward Euler and that the amplitude of the vibration appears to be constant. The ability of the Symplectic Euler Method to conserve energy is probably the main reason why it has been used in several of the mass-spring systems built for sound synthesis, such as the CORDIS system [4–6] and the TAO system [2, 16]. The symplectic Euler is a first order method, meaning that its global truncation error (the cumulative error) is proportional to the time step h. In a previous paper [11] we explored the possibility of using higher order symplectic methods. We concluded that in cases where the mass-spring system is being used to simulate a continuous system such as a vibrating string or two dimensional membrane, the extra accuracy of the higher order method is not worth the increase in computational cost. When a small number of masses are used to model a continuous system, the resulting mathematical model is not very accurate, and increasing the accuracy of the numerical method does not noticeably improve the sound of the simulation. If a large number of masses are used to model the continuous system, the model is much more accurate, but the sound produced will contain high-frequency components. In order to keep the system stable and to avoid aliasing, the sample rate has to be increased for both the symplectic Euler and the higher order methods. At a high sample rate the symplectic Euler method is quite accurate for the low and medium frequency components of the sound, and so there is still no appreciable difference in the sound of the simulation between the first-order symplectic Euler and higher order symplectic methods. This makes the symplectic Euler method a good choice for most sound synthesis applications using mass-spring systems. But it is still important to know the accuracy and stability limits of the method in order to set the sample rate of the simulation and resolve problems in situations where the sound produced is not as expected.

2.2. Mass-Spring Discretization. We now use the symplectic Euler method to discretize the mass-spring model.

Mass Element. We can derive the behavior of a mass from Newton's 2nd law:

$$F(t) = ma(t), \tag{8}$$

where $F(t)$ is the force acting on the mass at time t, m is the mass, and $a(t)$ is the acceleration of the mass at time t. Since acceleration is the derivative of the velocity we can write (8) as

$$F(t) = mv'(t). \tag{9}$$

We then have a system of two first-order differential equations:

$$\begin{pmatrix} x(t) \\ v(t) \end{pmatrix}' = \begin{pmatrix} v(t) \\ \dfrac{F(t)}{m} \end{pmatrix}. \tag{10}$$

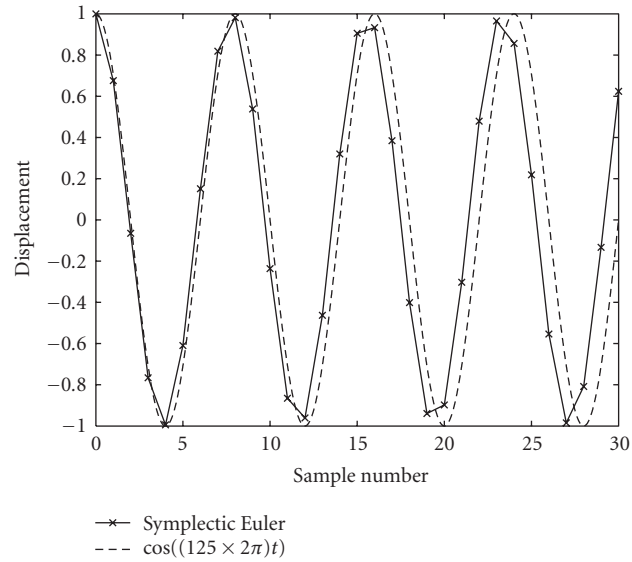

Figure 5: Symplectic Euler approximation of the undamped mass-spring system ($h = .001$).

We then use the symplectic Euler method to discretize these equations. Substituting $a(t) = F(t)/m$ in (7) gives us

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ v_n \end{pmatrix} + h \begin{pmatrix} v_{n+1} \\ \dfrac{F_n}{m} \end{pmatrix}, \tag{11}$$

where x is the displacement, v is the velocity and F_n is the numerical approximation of the force acting on the mass at time step n.

Note that if F_{n+1} was used instead of F_n in (7) we would have the backward Euler approximation. This is the way this numerical method has been described in the sound synthesis literature [2–4]; as the backwards Euler method with the forces delayed by one time step, rather than the symplectic Euler method.

Spring Element. The equations for the spring are derived using Hooke's Law $F(t) = -kx(t)$, where k is a constant denoting the spring stiffness. We write them as

$$F_{a:n+1} = k(x_{b:n+1} - x_{a:n+1}), \tag{12}$$

$$F_{b:n+1} = -F_{a:n+1}. \tag{13}$$

Here we let $x_{a:n}$, and $x_{b:n}$ represent the distance from the equilibrium position of mass M_a and mass M_b at either end of the spring. We use $F_{a:n}$ to denote the force acting on mass M_a at one end of the spring at time step n. The force, $F_{b:n}$, acting on mass M_b at the other end of the spring is, according to Newton's third law, equal and opposite to $F_{a:n}$.

Damper Element. The damper element is used to represent viscous friction. This is the object's resistance to motion and is assumed to be proportional to the velocity. The formula for the damper is $F(t) = -Zv_r(t)$, where Z is a constant

denoting the coefficient of viscosity, $F(t)$ the force, and $v_r(t)$ the relative velocity of the two ends of the damper. This can be written as

$$\begin{aligned} F_{a:n+1} &= Z(v_{b:n+1} - v_{a:n+1}), \\ F_{b:n+1} &= -F_{a:n+1}, \end{aligned} \tag{14}$$

where F_a and F_b represent the forces acting on the masses at the ends of the damper, and v_a and v_b are the corresponding velocities.

2.3. Synthesizing Sound Using a Mass-Spring System. The mass-spring system works by discretizing the physical equations of each of the elements—the masses, springs, and dampers—of a mass-spring system such as the one shown in Figure 1. At each time step the sums of the forces acting on each of the masses are calculated. These sums consist of the forces of the springs and dampers directly connected to the mass and any external forces acting on the mass. The external forces are used to simulate various physical interactions with the instrument, such as plucking, hitting, and bowing. The forces are used to calculate the new positions and velocities of the masses. Once the new positions and velocities have been calculated, they are used to calculate the new forces acting on each mass. This cycle then repeats for the duration of the simulation. The position of one of the masses at each time step in the simulation is written to a sound file to represent the sound produced at this point in the simulated instrument. Alternatively, the vibrations of several masses can be summed together and written to the sound file.

Since each element interacts only with the elements connected to it, the number of calculations depends on the number of masses and average number of connections of each mass. For simple systems these calculations can be done in real time [17]; more complex systems must be run offline. More detailed accounts of the implementation of mass-spring systems can be found in Cadoz et al. [4], and Pearson [2].

3. Analysis of a Damped Mass-Spring System Systems with a Single Mass

In this section we look at the stability and accuracy of the symplectic Euler method when used to simulate a damped mass-spring system containing a single mass, a single spring, and a single damper. We start by finding the analytical solution of this system. We find the z-transform of this equation when it is approximated by the symplectic Euler method. We then find the damping and the frequency of the discrete mass-spring system represented by transformed equation. We also find the conditions under which this system is stable.

3.1. The Analytical Solution of the Single Mass Damped Mass-Spring System. Figure 6 shows a damped mass-spring containing only one mass, M_1, one spring, S_1, and one damper, D_1.

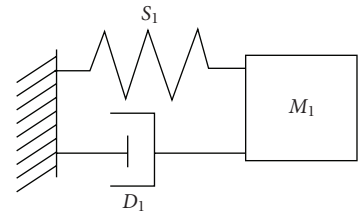

Figure 6: Damped mass-spring system.

The equation for this system is

$$mx''(t) + kx(t) + Zx'(t) = 0, \tag{15}$$

where $x(t)$ is the distance of the mass from its equilibrium position, m is the mass, k the spring stiffness, and Z the viscous damping coefficient.

We can solve this equation by finding the roots of the characteristic equation

$$r^2 + \frac{Z}{m}r + \frac{k}{m} = 0. \tag{16}$$

We use the substitutions $\gamma = Z/m$ and $\omega_0^2 = k/m$:

$$r^2 + \gamma r + \omega_0^2 = 0. \tag{17}$$

The roots of this characteristic equation are

$$r = \frac{-\gamma \pm \sqrt{\gamma^2 - 4\omega_0^2}}{2}. \tag{18}$$

For the system to vibrate we require that $\gamma^2 < 4\omega_0^2$, so

$$r = \frac{-\gamma \pm i\sqrt{4\omega_0^2 - \gamma^2}}{2}. \tag{19}$$

The condition dividing vibrating systems from those that do not vibrate occurs when

$$4\omega_0^2 = \gamma^2 \quad \text{or} \quad \gamma = 2\omega_0. \tag{20}$$

This value is known as *critical damping*.

The general solution of (15), when $\gamma^2 < 4\omega_0^2$, can be shown to be [15]

$$x(t) = e^{-\gamma t/2}\left(A\cos(\mu t) + B\sin(\mu t)\right), \tag{21}$$

where A and B are constants depending on the initial conditions. This can be written as [15]

$$x(t) = Re^{-\gamma t/2}\cos(\mu t + \phi), \tag{22}$$

where

$$\begin{aligned} R &= \sqrt{A^2 + B^2}, \\ \phi &= \tan^{-1}\left(\frac{B}{A}\right). \end{aligned} \tag{23}$$

This shows that the damped mass-spring system has a starting amplitude of R. This amplitude is being decreased by the term $e^{-\gamma t/2}$. The frequency (actually the *quasi-frequency* since the system is not strictly periodic) is $\mu = (1/2)\sqrt{4\omega_0^2 - \gamma^2}$. As the damping approaches zero this equation becomes $x(t) = R\cos(\omega_0 t + \phi)$.

3.2. The Damped Mass-Spring System Using the Symplectic Euler Method. We next examine the damping and frequency of the damped mass-spring system, when approximated by the symplectic Euler method, by using the z-transform.

The equation of the damped mass-spring system shown in Figure 6 is

$$mx''(t) + kx(t) + Zx'(t) = F_{\text{ext}}(t), \tag{24}$$

where F_{ext} is an external force acting on the mass. Using the substitutions from the previous section we can write (24) as

$$x''(t) = a(t) = -\omega_0^2 x(t) - \gamma v(t) + \frac{1}{m} F_{\text{ext}}(t). \tag{25}$$

We now discretize this equation using the symplectic Euler method by substituting (25) in (7):

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ v_n \end{pmatrix} + h \begin{pmatrix} v_{n+1} \\ -\omega_0^2 x_n - \gamma v_n + \dfrac{1}{m} F_{\text{ext:}n} \end{pmatrix}. \tag{26}$$

We should note that (26) can also be derived by substituting the equations for the spring (12) and the damper (14) together with an external force in the equation for the mass (11). This is how mass-spring systems actually work: by calculating the new values of each mass, spring, and damper using the equations from Section 2.2 at each time step. We can write (26) as a scalar equation by substituting the second line in the first line:

$$x_{n+1} = x_n + h\left(v_n + h\left(-\omega_0^2 x_n - \gamma v_n + \frac{1}{m} F_{\text{ext:}n}\right)\right). \tag{27}$$

Since, from the first line of (26), $v_{n+1} = 1/h(x_{n+1} - x_n)$, we can write v_n as $1/h(x_n - x_{n-1})$. Using this substitution in (27) gives us

$$\begin{aligned} x_{n+1} = x_n + h\Bigg(&\frac{1}{h}(x_n - x_{n-1}) \\ &+ h\left(-\omega_0^2 x_n - \gamma \frac{1}{h}(x_n - x_{n-1}) + \frac{1}{m} F_{\text{ext:}n}\right)\Bigg), \end{aligned} \tag{28}$$

which simplifies to

$$\begin{aligned} &\frac{1}{h^2}(x_{n+1} - 2x_n + x_{n-1}) \\ &\quad = -\omega_0^2 x_n - \gamma \frac{1}{h}(x_n - x_{n-1}) + \frac{1}{m} F_{\text{ext:}n}. \end{aligned} \tag{29}$$

We can shift the time step back by one to get

$$\begin{aligned} &\frac{1}{h^2}(x_n - 2x_{n-1} + x_{n-2}) \\ &\quad = -\omega_0^2 x_{n-1} - \frac{\gamma}{h}(x_{n-1} - x_{n-2}) + \frac{1}{m} F_{\text{ext:}n-1}. \end{aligned} \tag{30}$$

3.3. The **z***-Transform.* The z-transform takes signals from the time domain and transforms them to signals on the z-plane. The z-plane represents the signals in terms of amplitude growth (the growth or decay rate of the signal) and frequency, which are particularly useful in the analysis of musical signals. The z-transform is defined [18]:

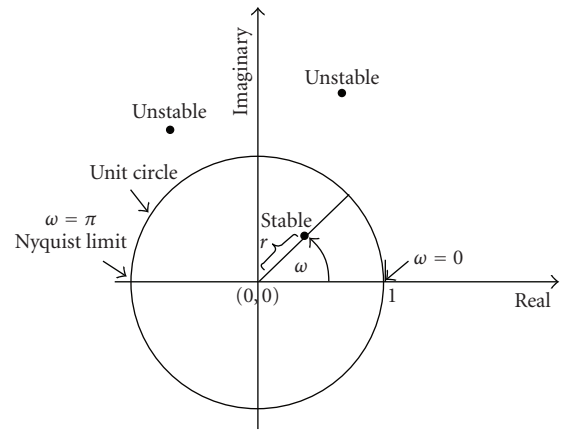

Figure 7: The z-plane.

$$\hat{X}(z) = \sum_{m=-\infty}^{\infty} x(m) z^{-m}, \tag{31}$$

where $z = re^{j\omega}$. The z-transform of a system can be represented on the *z-plane*, which is shown in Figure 7. The z-plane uses polar coordinates with $|z| = r$ being the distance from the origin and ω the angle. The amplitude growth or decay of the signal is represented by r and the frequency by ω. The frequency varies from 0 on the positive real axis, to π radians per sample on the negative real axis. This frequency, also known as the *Nyquist limit*, is the maximum frequency a discrete system can have. If the frequency goes above the Nyquist limit, it becomes indistinguishable from a frequency less than the Nyquist limit and the resulting frequency will be perceived as the lower of the two frequencies. This is known as *aliasing*, and causes inaccuracies in synthesized sound. The *transfer function* of a system is defined as the z-transform of its output divided by the z-transform of its input. The *poles* of a system are defined as the roots of the denominator of its transfer function. The poles are the *normal modes* or the *natural frequencies* of the system. A system is called stable if, when the input is absolutely summable, the output is absolutely summable. The system is stable on the z-plane if all its poles lie inside the unit circle [18] and marginally stable if it has a pole on the unit circle, but no pole outside the unit circle. A marginally stable system has a bounded output in some conditions, such as when the system has no input; but oscillations in a marginally stable system do not die away but persist indefinitely [19]. If any poles are outside the unit circle, the system is unstable. An important feature of the z-transform is that multiplying the z-transform of a signal by z^{-N} is the same as delaying the signal by N time steps [20], that is,

$$\text{if } y(n) = x(n - N) \text{ then } \hat{Y}(z) = z^{-N}\hat{X}(z). \tag{32}$$

3.4. Using the **z***-Transform to Find the Poles of the System.* Using the property of (32), the z-transform of (30) is

$$\begin{aligned}
&\frac{1}{h^2}\left(\hat{X}(z) - 2\hat{X}(z)z^{-1} + \hat{X}(z)z^{-2}\right) \\
&\quad = -\omega_0^2\hat{X}(z)z^{-1} - \frac{\gamma}{h}\left(\hat{X}(z)z^{-1} - \hat{X}(z)z^{-2}\right) + \frac{1}{m}\hat{F}_{\text{ext}}(z)z^{-1}, \\
&\hat{X}(z)\left(\frac{1}{h^2} + \left(\omega_0^2 + \frac{\gamma}{h} - \frac{2}{h^2}\right)z^{-1} + \left(\frac{1}{h^2} - \frac{\gamma}{h}\right)z^{-2}\right) \\
&\quad = \frac{1}{m}\hat{F}_{\text{ext}}(z)z^{-1},
\end{aligned} \tag{33}$$

and the transfer function is

$$\begin{aligned}
\hat{H}(z) &= \frac{\hat{X}(z)}{\hat{F}_{\text{ext}}(z)} \\
&= \frac{z^{-1}/m}{1/h^2 + (\omega_0^2 + \gamma/h - 2/h^2)z^{-1} + (1/h^2 - \gamma/h)z^{-2}} \\
&= \frac{(h^2/m)z^{-1}}{1 + \left((\omega_0 h)^2 + \gamma h - 2\right)z^{-1} + (1 - \gamma h)z^{-2}}.
\end{aligned} \tag{34}$$

The poles of the transfer function occur when

$$\begin{aligned}
1 + \left((\omega_0 h)^2 + \gamma h - 2\right)z^{-1} + (1 - \gamma h)z^{-2} &= 0, \\
z^2 + \left((\omega_0 h)^2 + \gamma h - 2\right)z + (1 - \gamma h) &= 0.
\end{aligned} \tag{35}$$

The roots of (35) are

$$z = \frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h \pm \sqrt{\left((\omega_0 h)^2 + \gamma h - 2\right)^2 - 4(1 - \gamma h)}\right) \tag{36}$$

$$= \frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h \pm \omega_0 h\sqrt{(\omega_0 h)^2 + 2\gamma h + \left(\frac{\gamma}{\omega_0}\right)^2 - 4}\right). \tag{37}$$

3.5. Using the Poles to Analyze the Mass-Spring System. The poles represent the *modes* or *natural frequencies* of the system. If the poles are complex, the discrete system will vibrate at a frequency in the range (0π) radians per sample. In this section we first determine what the conditions are for complex poles. We then find an equation for the damping of the discrete system when it has complex poles. The damping is determined by the distance of the pole from the origin on the z-plane. We then determine the frequency of the discrete system by finding the angle of the poles on the z-plane.

If the poles are real, then the frequency of the discrete system is either zero, if the larger magnitude pole is on the positive real axis, or π radians per sample, if the larger magnitude pole is on the negative real axis. We find the conditions determining whether the larger pole is positive or negative. We also find conditions for a pole being outside the unit circle, which will make the discrete system unstable.

We consider 2 cases in (37): (1) z is complex and (2) z is real.

3.5.1. Case (**1**)*—***z** *is Complex.* We look at the case in which z contains an imaginary component, that is, when

$$\left((\omega_0 h)^2 + 2\gamma h + \left(\frac{\gamma}{\omega_0}\right)^2\right) < 4. \tag{38}$$

We first calculate the range of values for ω_0 of this case. If

$$\begin{aligned}
(\omega_0 h)^2 + 2\gamma h + \left(\frac{\gamma}{\omega_0}\right)^2 &= 4 \\
\text{then } \omega_0^4 + \left(2\frac{\gamma}{h} - \frac{4}{h^2}\right)\omega_0^2 + \left(\frac{\gamma}{h}\right)^2 &= 0
\end{aligned} \tag{39}$$

by multiplying both sides by ω_0^2/h^2 and rearranging. The roots of this equation are

$$\omega_0^2 = \frac{1}{2}\left(\frac{4}{h^2} - \frac{2\gamma}{h} \pm \sqrt{\left(2\frac{\gamma}{h} - \frac{4}{h^2}\right)^2 - 4\left(\frac{\gamma}{h}\right)^2}\right). \tag{40}$$

This simplifies to

$$\omega_0^2 = \frac{2}{h^2} - \frac{\gamma}{h} \pm 2\sqrt{\frac{1}{h^4} - \frac{\gamma}{h^3}}. \tag{41}$$

If z is complex, ω_0^2 falls between these two roots.

We now look at the damping for case (1) (z is complex). In this case (37) becomes

$$z = \frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h \pm i\omega_0 h\sqrt{4 - (\omega_0 h)^2 - 2\gamma h - \left(\frac{\gamma}{\omega_0}\right)^2}\right). \tag{42}$$

We can calculate the length of z using the imaginary and real parts:

$$\begin{aligned}
|z| &= \sqrt{re(z)^2 + \text{imag}(z)^2} \\
&= \frac{1}{2}\sqrt{\left(2 - (\omega_0 h)^2 - \gamma h\right)^2 + (\mathfrak{G})^2},
\end{aligned} \tag{43}$$

where $\mathfrak{G}$ denotes $\omega_0 h\sqrt{4 - (\omega_0 h)^2 - 2\gamma h - (\gamma/\omega_0)^2}$. This simplifies to

$$|z| = \sqrt{1 - \gamma h}. \tag{44}$$

We see from (44) that, for this case, the length of z does not depend on the frequency.

Figure 8 shows an example of the poles on the z-plane of the symplectic Euler approximation of the damped mass-spring equation. The value for used for γh is .04. Root 1 and root 2 are the two roots of (37): root 1 has the surd added, while in root 2 it is subtracted. The arrows show how these roots—the poles of the system—move as $\omega_0 h$ varies from zero to above π radians per sample. The poles start on the positive real axis where the frequency is zero. While z is complex, each pole traces a semicircle, root 1—where the surd is added—is the upper semi-circle and root 2—where the surd is subtracted—is the lower semi-circle. If ω_0

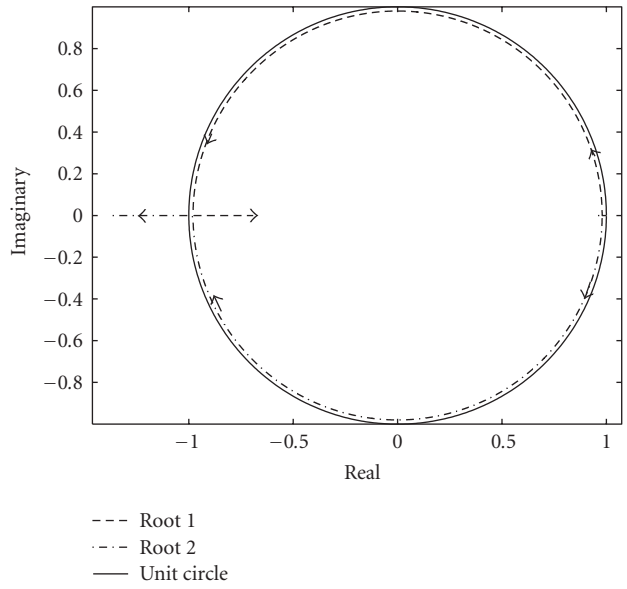

Figure 8: Damped mass-spring system using symplectic Euler-poles on z-plane with $\gamma h = .04$.

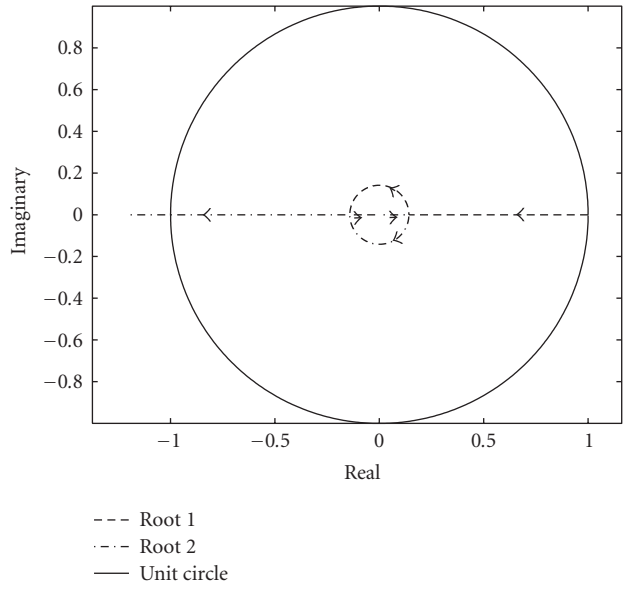

Figure 9: Damped mass-spring system using symplectic Euler-poles on z-plane with $\gamma h = .98$.

becomes large enough, the solutions of (37) become real again—but now they are on the negative real axis. Root 1 moves toward the origin, while root 2 becomes increasingly negative. When root 2 moves outside the unit circle, the system becomes unstable. Figure 9 shows the same plot with γh at .98. The radius of the circular region, where z is complex, is now very small.

The radius of the circle containing the poles is $\sqrt{1-\gamma h}$, which depends on both γ and the time step h. If the damping or the time step increase, the radius of the circle will become smaller. If the system is undamped (i.e., $\gamma = 0$), the circle will have a radius of one. This means that for the undamped system there is no numerical damping. This is consistent with the fact that the symplectic Euler method conserves energy.

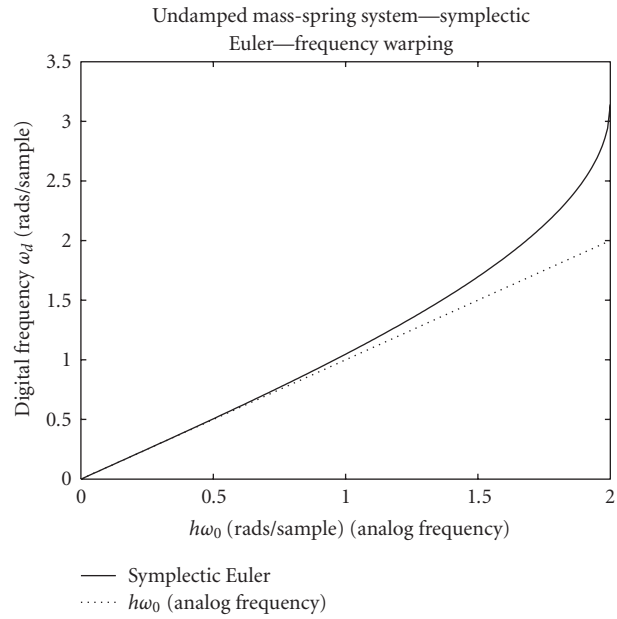

Figure 10: Frequency warping for undamped system.

We next look at the frequency warping when z is complex. From (42), we can calculate the frequency using the real and imaginary components. Using ω_d to denote the actual frequency obtained using the symplectic Euler,

$$\begin{aligned}\omega_d &= \tan^{-1}\left(\frac{\text{imaginary}(z)}{\text{real}(z)}\right)\\ &= \tan^{-1}\left(\frac{\omega_0 h\sqrt{4-(\omega_0 h)^2-2\gamma h-(\gamma/\omega_0)^2}}{2-(\omega_0 h)^2-\gamma h}\right).\end{aligned} \tag{45}$$

Figure 10 shows the frequency warping for the undamped mass-spring system. Note that for the undamped system the low frequencies are very accurate, and the range of $\omega_0 h$, where z is complex, goes from 0 to 2 radians per sample. As $\omega_0 h$ increases, the digital frequency (the actual frequency produced by the symplectic Euler approximation) becomes increasingly warped upward. When $\omega_0 h$ has reached 2 radians per sample, the digital frequency is π radians per sample—the Nyquist limit.

Figure 11 shows an example of the frequency warping of the symplectic Euler method when the damping is quite high: $h = .001$, $\gamma = 500$, and $\gamma h = .5$. The analog frequency is calculated as $\mu = (1/2)\sqrt{4\omega_0^2 - \gamma^2}$, and is slightly lower than ω_0. Note that the digital frequency has reached the Nyquist limit of π radians per sample at around $\omega_0 h = 1.7$ radians per sample. This is consistent with (41), which gives .293 as the lower limit and 1.707 as the upper limit for z being complex. As γh increases, the range where z is complex decreases and the frequency warping becomes more pronounced.

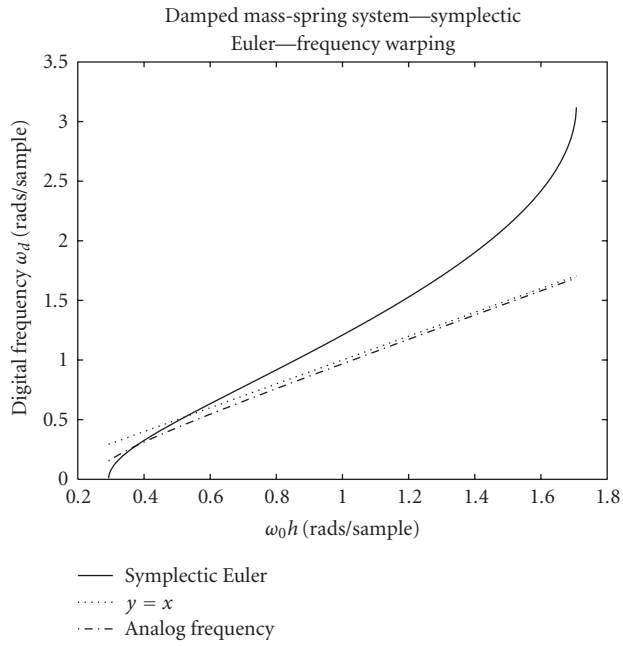

Figure 11: Frequency for damped mass-spring system, $\gamma h = .5$.

3.5.2. Case (**2**)—**z** *is Real.* Equation (41) gives us 2 conditions for z to be real. Equation (36) yields 2 roots, which are the two poles of the system. The pole with the larger magnitude will dominate the system. If the pole with the larger magnitude is real and positive, the frequency of the system is zero. If it is real and negative, the system will vibrate at π radians per sample.

We show that, if ω_0^2 is greater than the root of (41) in which the surd is added, then (36) will have its largest magnitude root less than zero. Since the surd being added or subtracted in (36) is the same, what we want to show is simply that

$$\begin{aligned}&\omega_0^2 > \left(\frac{2}{h^2} - \frac{\gamma}{h} + 2\sqrt{\frac{1}{h^4} - \frac{\gamma}{h^3}}\right)\\ &\text{implies } \left(\frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h\right)\right) < 0.\end{aligned} \tag{46}$$

We also assume that γh is less than 1. The case where $\gamma h \geq 1$ is considered later. We can write the left side of (46) as

$$(\omega_0 h)^2 > \left(2 - \gamma h + 2\sqrt{1 - \gamma h}\right). \tag{47}$$

From these facts we can derive that

$$\begin{aligned}&\left(\frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h\right)\right)\\ &\quad < \frac{1}{2}\left(2 - \left(2 - \gamma h + 2\sqrt{1 - \gamma h}\right) - \gamma h\right)\\ &\quad = -\sqrt{1 - \gamma h} < 0 \quad (\text{assuming } \gamma h < 1).\end{aligned} \tag{48}$$

Since the pole with the larger magnitude is on the negative real axis of the z-plane, the frequency is π radians per sample.

Similarly, if ω_0^2 is less than the root of (41) in which the surd is subtracted, that is, if

$$(\omega_0 h)^2 < \left(2 - \gamma h - 2\sqrt{1 - \gamma h}\right), \tag{49}$$

then

$$\begin{aligned}&\left(\frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h\right)\right)\\ &\quad > \frac{1}{2}\left(2 - \left(2 - \gamma h - 2\sqrt{1 - \gamma h}\right) - \gamma h\right)\\ &\quad = \sqrt{1 - \gamma h} > 0 \quad (\text{assuming } \gamma h < 1).\end{aligned} \tag{50}$$

So in this case, since the pole with the larger magnitude is on the positive real axis of the z-plane, the frequency is zero.

As the leftmost pole approaches -1, the damping becomes smaller. The system will become unstable when z becomes less than -1. This happens when from (37)

$$\begin{aligned}&\frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h - \omega_0 h\sqrt{(\omega_0 h)^2 + 2\gamma h + \left(\frac{\gamma}{\omega_0}\right)^2 - 4}\right) = -1,\\ &4 - (\omega_0 h)^2 - \gamma h = \omega_0 h\sqrt{(\omega_0 h)^2 + 2\gamma h + \left(\frac{\gamma}{\omega_0}\right)^2 - 4}.\end{aligned} \tag{51}$$

Squaring both sides and simplifying results in

$$(\omega_0 h)^2 + 2\gamma h - 4 = 0. \tag{52}$$

For a fixed time step, h, we can solve for ω_0:

$$\begin{aligned}\omega_0^2 &= \frac{4 - 2\gamma h}{h^2},\\ \omega_0 &= \frac{1}{h}\sqrt{4 - 2\gamma h}.\end{aligned} \tag{53}$$

So for stability we require that

$$\omega_0 \leq \frac{1}{h}\sqrt{4 - 2\gamma h}. \tag{54}$$

As the value of γh increases, the circular region on the z-plane where z is complex becomes smaller, as can be seen in Figure 9 where $\gamma h = .98$. When $\gamma h = 1.0$ the circular region disappears. For $\gamma h \geq 1.0$, there are only two possible digital frequencies: zero and π radians per sample. If the pole with the larger magnitude is negative, the digital frequency will be π radians per sample, otherwise it will be zero. From (37), the pole with the larger magnitude is negative when

$$2 - (\omega_0 h)^2 - \gamma h < 0, \tag{55}$$

so, when $\gamma h \geq 1.0$, the curve dividing systems that do not vibrate and those that vibrate at π radians per sample is

$$2 - (\omega_0 h)^2 - \gamma h = 0. \tag{56}$$

3.6. Regions of the **s***-Plane.* We are using a mass-spring system as a way of mathematically modelling a vibrating physical system. We then use numerical methods to approximate this mathematical model and implement a simulation of the system on a computer. There are 2 sources of error.

(1) Discrepancies between the vibrating system and the mathematical model.

(2) Discrepancies between the mathematical model and the computer simulation.

In this paper, our concern is with the second error. We want to know, given the mathematical model, how accurate is the computer simulation? The mathematical model, which is a continuous model, can be analyzed on the s-plane. Analogous to the z-plane, the s-plane represents the poles of the analog (continuous) system where $s = \sigma + \mu j$. The s-plane uses rectangular coordinates with the horizontal axis—the real axis—representing the decay rate and the vertical axis—the imaginary axis—representing the frequency. The decay rate is denoted by $\sigma = -\gamma/2$ and the frequency by $\mu = (1/2)\sqrt{4\omega_0^2 - \gamma^2}$. The mathematical model, which we refer to as the analog system, is stable if all its poles are on the left hand side of the s-plane. Ideally, all stable analog systems would result in stable computer simulations. However, this is not necessarily the case, since numerical methods are approximations. In this section, we show which systems having stable mathematical models will be stable when simulated using the symplectic Euler method. We do this by showing graphically which parts of the left hand side of the s-plane will be mapped to stable systems on the z-plane, that is, which poles on the s-plane will, when the system is simulated using the symplectic Euler method, result in poles within the unit circle on the z-plane.

We can find the region on the s-plane that maps to stable poles on the z-plane by using (54). In this case, however, we need it in terms of μ and σ. First we find ω_0^2 in terms of μ and σ:

$$\begin{aligned} \mu &= \frac{1}{2}\sqrt{4\omega_0^2 - \gamma^2}, \\ \mu^2 &= \frac{1}{4}\left(4\omega_0^2 - 4\sigma^2\right), \\ \omega_0^2 &= \mu^2 + \sigma^2. \end{aligned} \tag{57}$$

We then substitute (57) in (53):

$$\begin{aligned} \mu^2 + \sigma^2 &= \frac{1}{h^2}(4 - 2\gamma h), \\ \mu &= \pm\frac{1}{h}\sqrt{4 + 4\sigma h - (\sigma h)^2}. \end{aligned} \tag{58}$$

The system is then stable if

$$-\frac{1}{h}\sqrt{4 + 4\sigma h - (\sigma h)^2} \le \mu \le \frac{1}{h}\sqrt{4 + 4\sigma h - (\sigma h)^2}. \tag{59}$$

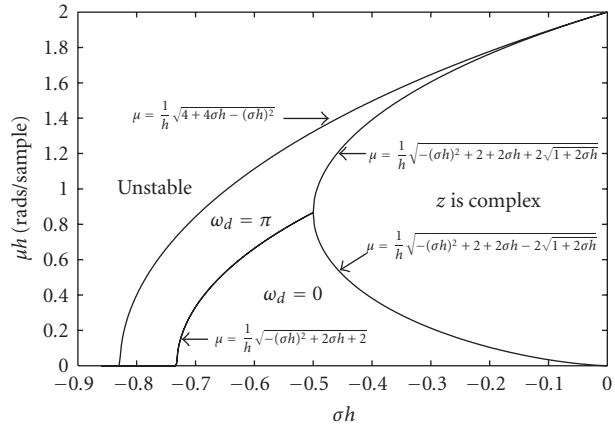

Figure 12: Regions of the s-plane.

We can also solve (41) and (56) in terms of μ and σ. Equation (41) divides poles that are complex from those that are real. Substituting (57) into (41) we get

$$\begin{aligned} \mu^2 + \sigma^2 &= \frac{2}{h^2} - \frac{\gamma}{h} \pm 2\sqrt{\frac{1}{h^4} - \frac{\gamma}{h^3}}, \\ \mu &= \pm\frac{1}{h}\sqrt{-(\sigma h)^2 + 2 + 2\sigma h \pm 2\sqrt{1 + 2\sigma h}}. \end{aligned} \tag{60}$$

Similarly, solving (56)—the curve dividing vibrating poles from nonvibrating poles—for μ in terms of σ results in

$$\mu = \pm\frac{1}{h}\sqrt{-(\sigma h)^2 + 2\sigma h + 2}. \tag{61}$$

Using these equations, we can divide the left half on the s-plane into sections. These regions represent the qualitative properties that any pole on the s-plane within the region will have when the systems is approximated with the symplectic Euler method. Figure 12 shows the regions of the s-plane for positive frequencies. The negative frequencies are mirror images of the positive ones. If any of the poles of the analog system lie within the unstable region, the discrete system resulting from the symplectic Euler method will be unstable.

3.7. The Accuracy of the Damping. In this section we determine the accuracy of the damping of a mass-spring system discretized by the symplectic Euler method. We determine the equation for the digital damping and examine some of its points of interest. We find that it has a sharp change of direction when it moves from one region (from Figure 12) to another. It turns out that in some places, counter-intuitively, increasing the damping coefficient γ actually decreases the digital damping. We also find that the equation for digital damping has a singular point where the damping becomes infinite, and we determine exactly where that point is.

From (22), the damping of the analog system is $e^{-\gamma t/2}$. On the s-plane this is $e^{\sigma t}$. The damping of the discrete system is r^n where $r = |z|$. How do these two values compare? If we sample the continuous damping at each time step, we have $e^{\sigma n h}$ as the samples of the analog damping, where n is the

sample number and h the length of the time step. The discrete damping is

$$r^n = e^{n\ln r} = e^{\sigma_d nh}, \tag{62}$$

where σ_d denotes the damping of the discrete system. So,

$$\sigma_d = \frac{1}{h}\ln r = \frac{1}{h}\ln|z|. \tag{63}$$

Using the value of z from (37) we can compare the analog to the digital damping. Figure 13 shows the damping of the symplectic Euler method when $\mu h = .1$. When the analog damping is small (i.e., σ is near zero), it is quite accurate. Note that, since the accuracy of the damping depends on both σ and h, we can increase the accuracy of the damping by decreasing the time step h, because, for a fixed value of σ, decreasing h will move the digital damping of the system toward the right side of the figure where the value of $\sigma_d h$ is very close to σh. There are two points when the function has a sharp corner: the first is when z leaves the region where it is complex and enters the region where the frequency is zero; the second is where z enters the region where the system vibrates at π radians per sample. Once z is in the region where the digital frequency is π radians per sample, the digital damping decreases (i.e., σ_d approaches 0) and the system becomes unstable when the digital damping is greater than zero. The positions of the vertical lines marking where the digital frequency becomes zero and π radians per sample are found by solving (60) and (61), respectively, for σh in terms of μh, and finding the value of σh, given the known value of μh. For (61) this works out to

$$\sigma h = 1 \pm \sqrt{3 - (\mu h)^2}. \tag{64}$$

For (60), solving for σh results in a very long fourth degree equation.

Solving for σh when $\mu h = .1$ results in the value $\sigma h = -0.1954$ for the line at which z leaves the region where it is complex and enters the region where the digital frequency is zero, and $\sigma h = -0.7292$ for the line dividing systems that have digital frequencies of zero from those having digital frequencies of π radians per sample. We can also see these values by drawing a horizontal line, for this example at $\mu h = .1$, in Figure 12. We see that the points at which the horizontal line enters the regions $\omega_d = 0$ and $\omega_d = \pi$ match those of Figure 13. Figure 14 shows the digital damping when $\mu h = .8$. Solving for σh when $\mu h = .8$ results in the value $\sigma h = -0.4983$ for the line at which z leaves the region where it is complex and enters the region where the digital frequency is zero, and $\sigma h = -0.5362$ for the line dividing systems that have digital frequencies of zero from those having digital frequencies of π radians per sample.

Figure 15 shows the digital damping when $\mu h = .87$. Solving for σh when $\mu h = .87$ results in the value $\sigma h = -0.4977$ for the line at which z leaves the region where it is complex and enters the region where the digital frequency is π radians per sample.

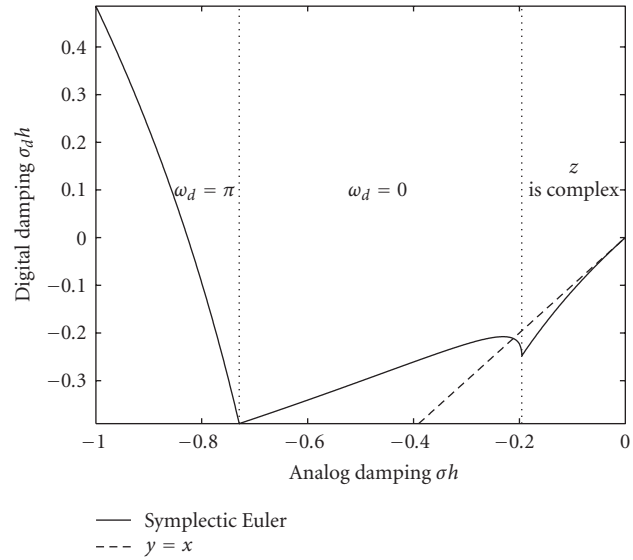

Figure 13: Digital damping versus analog damping—$\mu h = .1$.

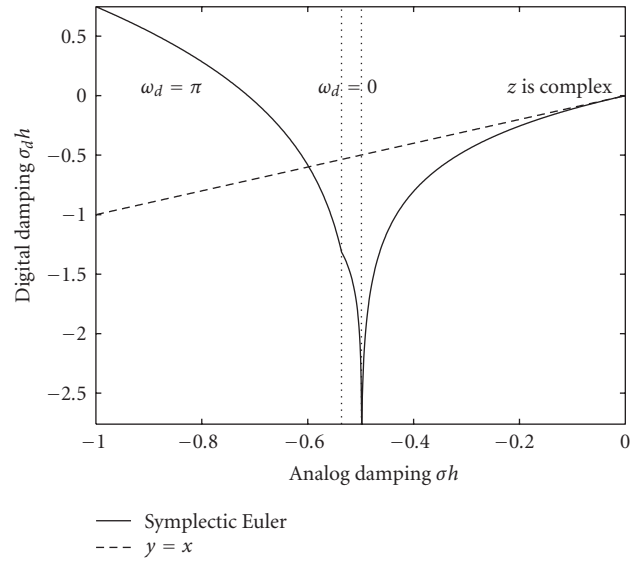

Figure 14: Digital damping versus analog damping—$\mu h = .8$.

From (63) we see that when $r = |z|$ approaches zero, the digital damping approaches ∞. This happens when from (36)

$$\begin{aligned}
z &= \frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h \pm \sqrt{\left((\omega_0 h)^2 + \gamma h - 2\right)^2 - 4(1 - \gamma h)}\right) = 0,\\
(\omega_0 h)^2 + \gamma h - 2 &= \pm\sqrt{\left((\omega_0 h)^2 + \gamma h - 2\right)^2 - 4(1 - \gamma h)},\\
\left((\omega_0 h)^2 + \gamma h - 2\right)^2 &= \left((\omega_0 h)^2 + \gamma h - 2\right)^2 - 4(1 - \gamma h),\\
1 - \gamma h &= 0,\\
\gamma h &= 1.
\end{aligned} \tag{65}$$

If $\gamma h = 1$ and $z = 0$, then

$$z = \frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h \pm \sqrt{\left((\omega_0 h)^2 + \gamma h - 2\right)^2 - 4(1 - \gamma h)}\right) = 0,$$

$$1 - (\omega_0 h)^2 \pm \sqrt{\left((\omega_0 h)^2 - 1\right)^2} = 0,$$

$$1 - (\omega_0 h)^2 \pm \left((\omega_0 h)^2 - 1\right) = 0,$$

$$0 = 0 \quad \text{or} \quad 2 - 2(\omega_0 h)^2 = 0,$$

$$(\omega_0 h)^2 = 1. \tag{66}$$

If $(\omega_0 h)^2 = 1$ then from (57)

$$(\mu^2 + \sigma^2) h^2 = 1. \tag{67}$$

If $\gamma h = 1$, then $\sigma h = -.5$, so

$$(\mu h)^2 = 1 - .5^2,$$

$$\mu h = \sqrt{\frac{3}{4}} = \frac{\sqrt{3}}{2}. \tag{68}$$

As μh approaches $\sqrt{3}/2 \approx 0.866$, the digital damping becomes increasingly large when σh approaches -0.5. We can observe this in Figures 14 and 15 by the large dip at $\sigma h = -0.5$. We can also observe this in Figure 17 in the next section. When μh is greater than $\sqrt{3}/2$, the region where the digital frequency is zero disappears, as can be seen in Figure 15.

3.8. Isofrequencies and Isodamping of the Symplectic Euler. Figure 16 graphically represents how the symplectic Euler affects the frequency of a mass-spring system. The horizontal lines have the same digital frequencies (isofrequencies). For example, the label "$\omega_d h = 2$" on the right side of Figure 16, marks the line where the digital frequency is 2 radians per sample. The right endpoint of the line occurs when the damping is zero. At this point the analog frequency (using the scale on the left side of the figure) is around 1.68 radians per sample. As we follow this line to the left it gets lower and lower on the graph, indicating that the frequency warping is increasing as the damping increases. When the damping is $\sigma h = -0.4$, the analog frequency is around 1.2 radians per sample, for a digital frequency of $\omega_d h = 2$. By plotting an s-plane pole on this graph, we can find the effect the symplectic Euler will have on the frequency by noting which isofrequency line the pole is near. Figure 16 was created by solving (45), which calculates the frequency warping, for $\omega_0 h$. This allows us to calculate the value for $\omega_0 h$ for fixed values of the digital frequency $\omega_d h$ and damping σh. We can then convert $\omega_0 h$ to μh, the analog frequency, using (57) and plot the point on the graph. This is repeated for each value of σh going from $\sigma h = 0$ to $\sigma h = -0.5$ in small increments, resulting in one isofrequency line. The process is repeated for each isofrequency.

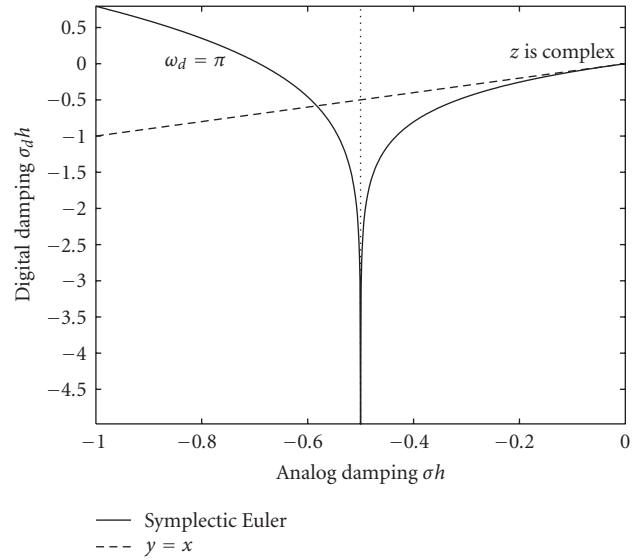

Figure 15: Digital damping versus analog damping— $\mu h = .87$.

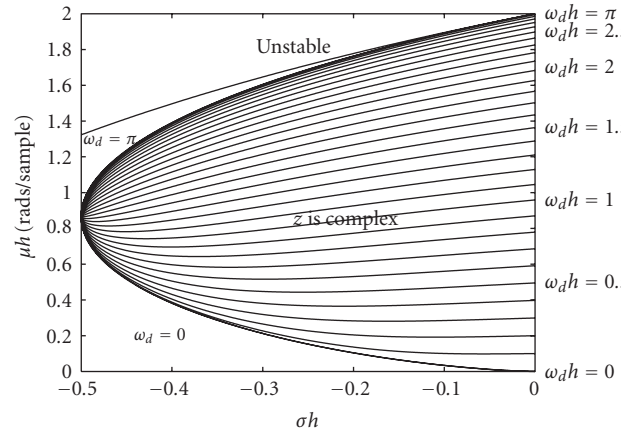

Figure 16: Isofrequencies of symplectic Euler.

Figure 17 shows how the symplectic Euler affects the damping of a mass-spring system. Note that, in the area where z is complex, the isodamping has straight vertical lines. This is because, in this region, the digital damping does not depend on the frequency, as shown by (44): $|z| = \sqrt{1 - \gamma h}$. We can also see that as we approach the region of instability, the digital damping decreases to zero. Figure 17 was created in a similar manner to Figure 16. This time (63) is solved for σh for fixed values of $\sigma_d h$ and μh. Each isodamping line is plotted by holding σ_d at the desired value while increasing μh in small increments.

3.9. Testing the Theory. We can test our theoretical results by comparing samples from a cosine wave using the digital frequency and damping calculated by (45) and (44) to the actual output produced by software running the symplectic

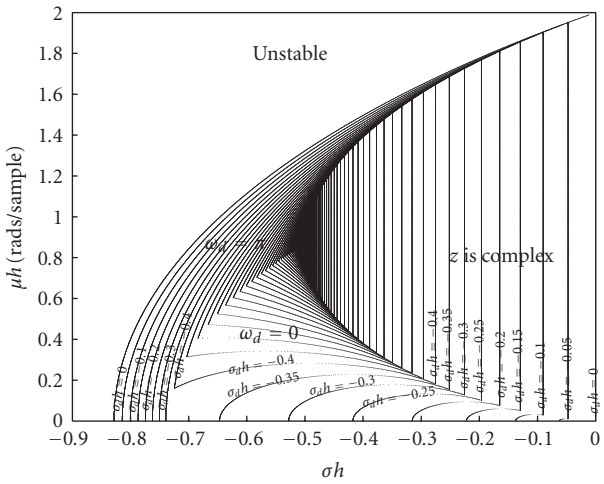

Figure 17: Isodamping of symplectic Euler.

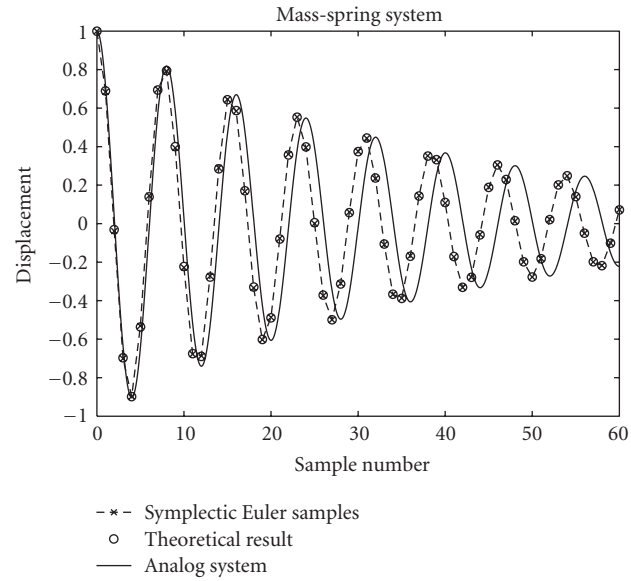

Figure 18: Testing the theoretical model against the actual results.

Euler method. For our test we use the constants

$$\omega_0 = 125 \times 2\pi, \qquad \gamma = 50, \qquad h = 1/1000, \qquad x(0) = 1, \qquad v(0) = 0. \tag{69}$$

We use $x(0)$ for the initial displacement and $v(0)$ for the initial velocity. Using the equations from Section 3.1 we calculate the analytical solution as

$$x(t) = Re^{-\gamma t/2}\cos(\mu t + \phi),$$
$$\text{with } R = 1.00050699121789, \quad \mu = 785.0001752025822, \quad \phi = 0.03183636632642. \tag{70}$$

For the theoretical results of the symplectic Euler method, we use (45) to calculate the digital frequency, ω_d, and (44) to calculate the digital damping, $r = |z|$. The equation for the theoretical results is

$$x_n = Rr^n\cos(\omega_d nh + \phi),$$
$$\text{with } r = 0.97467943448090, \quad \omega_d = 817.7132374981528, \quad \phi = 0.03183636632642, \quad R = 1.00050699121789. \tag{71}$$

Figure 18 shows a plot of the theoretical model of the symplectic Euler method, the solution of continuous system and actual samples produced by the symplectic Euler method using the equations from Section 2.2 which result in a system represented by (26). The external force in this example is set to zero. We see that the theoretical model's samples match almost exactly the samples produced by the actual symplectic Euler method (since the o's and the x's on the graph overlap), giving us confidence the theory is correct.

Figure 19 shows a second test, this time it uses the constants

$$\omega_0 = 1708, \qquad \gamma = 500, \qquad h = 1/1000, \qquad x(0) = 1, \qquad v(0) = 2324. \tag{72}$$

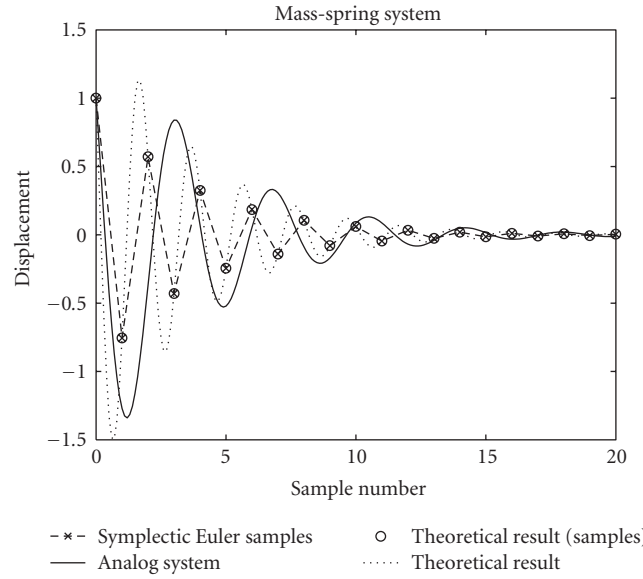

Figure 19: Testing the theoretical model against the actual results—actual frequency is π radians/sample.

For this simulation the digital frequency is π radians per sample, although the simulation is still stable. The dotted line in Figure 19 shows the equation for the theoretical results, (71), for a continuous time scale, that is, t instead of nh.

4. Generalizing to Mass-Spring Systems with Multiple Degrees of Freedom

So far, we have just analyzed mass-spring systems with a single mass. We now consider systems with multiple masses. Since each mass can move independently of the other masses, these systems are said to have multiple degrees of

freedom. We first look at the analytical solution of the general mass-spring system with n degrees of freedom using the method presented by Meirovitch [21] and then look at how discretization by the symplectic Euler method affects a mass-spring system with multiple masses.

4.1. Analytical Solution of Mass-Spring Systems. We can find the analytical solution of a mass-spring system by using the state-space method. The state space method uses a vector, $\mathbf{x}(t)$, of state variables. The equation describing the state variables of the state-space system is [22]

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{73}$$

where the vector $\mathbf{u}(t)$ is the input and A and B are matrices. The output is produced from the state variables and input by the equation

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t). \tag{74}$$

The state variables should contain the information needed to calculate the system's configuration at each point in time, so for a mass-spring system an obvious choice is the displacement and velocity of each of the masses. So $\mathbf{x}^T(t)$ is

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_n & x_1' & x_2' & \cdots & x_n' \end{pmatrix}, \tag{75}$$

where x_i represents the displacement of mass i and x_i' the velocity of mass i.

The first step is to determine the matrix A. Following Meirovitch [21], we create A in terms of the mass, stiffness, and damping matrices. The mass matrix, M, contains each of the masses along its diagonal

$$M = \begin{pmatrix} m_1 & 0 & 0 & \cdots & 0 \\ 0 & m_2 & 0 & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & 0 & m_n \end{pmatrix}. \tag{76}$$

The stiffness matrix, K, contains stiffness influence coefficients, k_{ij}, that are defined as the forces required for a unit displacement of mass i, with all other masses $j \neq i$ having a displacement of zero.

The damping matrix, Z, contains damping coefficients, Z_{ij}, that are defined as the forces required for a unit velocity of mass i to the right, with all other masses $j \neq i$ having a velocity of zero [23]. If we assume $\mathbf{u}(\mathbf{t}) = 0$ (i.e., no external force is acting on the system), we can then write (73) as

$$\frac{d}{dt}\mathbf{x}(t) = \left(\begin{array}{c|c} \mathbf{0} & I \\ \hline -M^{-1}K & -M^{-1}Z \end{array}\right)\mathbf{x}(t), \tag{77}$$

where the input, $\mathbf{u}(t)$, is zero and the matrix A is the matrix on the right side of the equation. This is the general form of matrix A for a state space mass-spring system. If there are n masses, A is an $2n \times 2n$ matrix, $\mathbf{0}$ is an $n \times n$ matrix of zeros, I is the $n \times n$ identity matrix, and $-M^{-1}K$ and $-M^{-1}Z$ are both $n \times n$ matrices.

For a mass-spring system the input $\mathbf{u}$ is an external force acting on each mass. If the system has no external force $\mathbf{u}$ is equal to zero. The matrix B is the $n \times n$ identity matrix. Since we want the state vector $\mathbf{x}$ as output, the matrix C is the $n \times n$ identity matrix and $D = \mathbf{0}$.

The solution to the state space system is [22]

$$\begin{aligned} \mathbf{x}(t) &= e^{tA}\mathbf{x}(0) + \int_0^t e^{t-\tau}B\mathbf{u}\,d\tau, \\ \mathbf{y}(t) &= C\left(e^{tA}\mathbf{x}(0) + \int_0^t e^{t-\tau}B\mathbf{u}\,d\tau\right) + D\mathbf{u}. \end{aligned} \tag{78}$$

The matrix exponential, e^{tA} is defined as

$$e^{tA} = I + \frac{t}{1!}A + \frac{t^2}{2!}A^2 \cdots + \frac{t^k}{k!}A^k \cdots. \tag{79}$$

If there is no external force (i.e., $\mathbf{u}(t)$ is equal to zero) and the matrix $D = 0$, then solution of the state space system simplifies to

$$\mathbf{y}(t) = Ce^{tA}\mathbf{x}(0). \tag{80}$$

More detailed information on the analytical solution of mass-spring systems can be found in the book by Meirovitch [21], which this section is based on.

4.2. Mass-Spring Systems with Multiple Masses Using the Symplectic Euler Method. The poles of the analog system on the s-plane are the eigenvalues of the system matrix (the matrix A described in the previous section). For each mass in the mass-spring system, we have a conjugate pair of poles. We can view numerical methods as mapping for the s-plane to z-plane, so if any pole on the s-plane is mapped outside the unit circle on the z-plane, the system will be unstable. We can determine if a mass-spring system with multiple masses will be stable when simulated with the symplectic Euler method by calculating the eigenvalues of the system matrix and testing each eigenvalue for stability using (59). The real part of the eigenvalue is σ and the imaginary part is μ. If all the eigenvalues are stable, the simulation of the system will be stable; otherwise it will be unstable. Equivalently, we can plot the poles against the regions of the s-plane as shown in Figure 12. If all the poles are in the stable regions, the simulation of the system will be stable.

4.3. Example—Finding Coefficients for a Vibrating String. We show how to use the results from the previous sections to to find the correct spring stiffness and damping coefficients for a vibrating string simulated with a mass-spring system using the symplectic Euler method. Figure 20 shows a simulated string constructed using N masses, each connected by a spring and damper. For an ideal continuous string, there are an infinite number of frequencies, each of which are integer multiples of the fundamental frequency [24], which is a function of the tension, the string length, and the mass

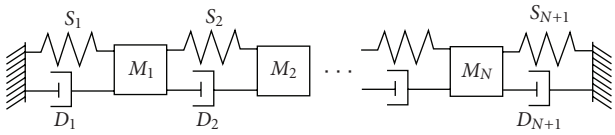

Figure 20: Damped mass-spring system with N masses.

per unit length. For example, if the fundamental frequency is 440 Hz, the frequencies are 440 Hz, 880 Hz, 1320 Hz, 1760 Hz, and so forth.

For a mass-spring system, the number of frequencies depends on the number of masses. A simulated string with N masses will have N frequencies [24]. As the number of masses in the mass-spring system becomes very large, the audible frequencies also approach integer multiples of the fundamental [24], but for smaller systems, the higher partials have lower frequencies than those of a continuous string.

We now consider an algorithm that, when given the sample time, h, the number of masses, N, the desired fundamental frequency, F, and the time constant, τ, calculates the values for the mass-spring system's coefficients: $\omega_{0\text{out}} = \sqrt{k/m}$ and $\gamma_{\text{out}} = Z/m$. The time constant for the fundamental frequency, τ, is the length of time it takes for the amplitude to decay to $1/e$ that of the starting amplitude. It is assumed that the mass, spring stiffness, and damping coefficients are the same for each mass, spring, and damper, respectively.

We start by calculating the value for the digital damping, σ_d, given the value of the time constant τ. The damping of the system is $e^{\sigma t}$, so from the definition of the time constant

$$\begin{aligned} e^{\sigma_d \tau} &= \frac{1}{e}, \\ \sigma_d &= \frac{1}{\tau}\ln\left(e^{-1}\right) = -\frac{1}{\tau}. \end{aligned} \tag{81}$$

This is the value for the digital damping that has the time constant τ. We want to find the σ_1, which after numerical damping, has the time constant τ. In other words we want to find σ_1 such that

$$D(\sigma_1) = \sigma_d, \tag{82}$$

where D is numerical damping function. To do this we use the inverse of the numerical damping function, D^{-1}, with

$$D^{-1}(\sigma_d) = \sigma_1. \tag{83}$$

The equation for numerical damping is $\sigma_d = (1/h)\ln r$ (63). When z is complex, $r = |z| = \sqrt{1-\gamma h}$ (44). Equating this with (81) and solving for γ gives us

$$\begin{aligned} \sigma_d &= \frac{1}{h}\ln\left(\sqrt{1-\gamma h}\right) = -\frac{1}{\tau}, \\ \gamma &= \frac{1}{h}\left(1 - e^{-2h/\tau}\right). \end{aligned} \tag{84}$$

Since $\sigma = -\gamma/2$, the value for σ_1 is

$$D^{-1}(\sigma_d) = -\frac{1}{2h}\left(1 - e^{-2h/\tau}\right) = \sigma_1. \tag{85}$$

This is the value for the damping of the lowest frequency, that after numerical damping will give the correct value for τ. That means that the real part of the eigenvalue with the lowest frequency should have this value. We create a function, $g(\gamma, \omega_0, N)$, that creates a system matrix with fixed values for N and ω_0, and the variable γ. It then calculates the eigenvalues of the system matrix and returns the real part of the eigenvalue with the lowest frequency. Since the damping values do not depend on ω_0, we can use a rough approximation for it. We can then solve the equation $g(\gamma_{\text{out}}, \omega_0, N) - \sigma_1 = 0$ numerically to find the value of γ_{out} where $g(\gamma_{\text{out}}, \omega_0, N) = \sigma_1$. We can use any zero finding method to solve this equation. The resulting value, γ_{out}, is the value we use to calculate the coefficient Z, where $Z = \gamma_{\text{out}} m$, for the damping coefficient of each damper.

We then use the same approach to find $\omega_{0\text{out}}$, the value used to set the spring stiffness coefficient, $k = \omega_{0\text{out}}^2 m$, for each spring. We first find the frequency, μ_1, for the imaginary part of the eigenvalue with the lowest frequency. We want this to be the value that, after frequency warping, results in the desired fundamental frequency, F. So

$$W(\mu_1, \gamma, h) = F, \qquad W^{-1}(F, \gamma, h) = \mu_1, \tag{86}$$

where W is the frequency warping function. To find W^{-1}, we solve the equation for frequency warping, (45), for ω_0. This works out to be a quadratic equation in $(\omega_0 h)^2$, with

$$\begin{aligned} (\omega_0 h)^2 &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = R_1, R_2, \text{ where} \\ a &= \tan^2(\omega_d) + 1, \quad b = 2\gamma h - 4 + \tan^2(\omega_d)(2\gamma h - 4), \\ c &= (\gamma h)^2 + \tan^2(\omega_d)\left(4 - 4\gamma h + (\gamma h)^2\right), \end{aligned} \tag{87}$$

where R_1 and R_2 are the two roots of the quadratic equation. We then use the substitution $\omega_0^2 = \mu^2 + \sigma^2$ (57) and solve for μ to get

$$\begin{aligned} (\mu h)^2 + (\sigma h)^2 &= R_1, R_2 \\ \mu &= \frac{\sqrt{R_1^2 - \sigma^2}}{h}, \frac{\sqrt{R_2^2 - \sigma^2}}{h} = \frac{\sqrt{R_2^2 - \gamma^2/4}}{h}, \frac{\sqrt{R_1^2 - \gamma^2/4}}{h}. \end{aligned} \tag{88}$$

If we calculate the derivative of (45), we find that it is always positive, showing that the frequency warping is an increasing function. This means that the inverse is also a monotonic function. Squaring, and then taking the square root has introduced extraneous roots. The correct function for the inverse frequency warping is

$$W^{-1}(\omega_d, \gamma) = \begin{cases} \omega_d h \le \dfrac{\pi}{2} & \dfrac{\sqrt{R_2^2 - \gamma^2/4}}{h} \\ \omega_d h > \dfrac{\pi}{2} & \dfrac{\sqrt{R_1^2 - \gamma^2/4}}{h}. \end{cases} \tag{89}$$

The value of μ_1, the imaginary part of the eigenvalue with the smallest frequency, is then $W^{-1}(F, \gamma, h)$, where F is the desired fundamental frequency and $\gamma = -2\sigma_1$.

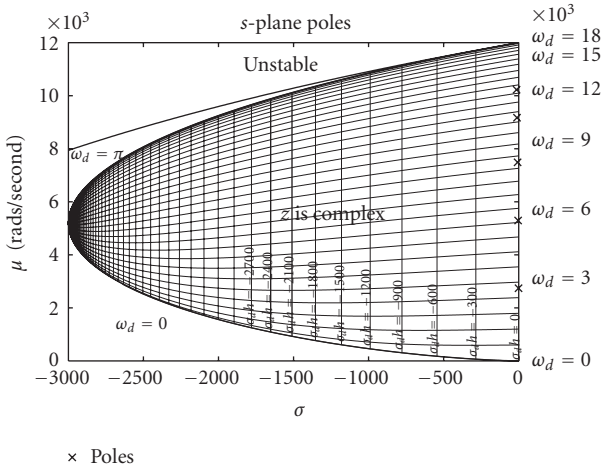

Figure 21: Poles of simulated string with 5 masses.

The last step is to find $\omega_{0\text{out}}$, the value of of ω_0 to use for each spring of the mass-spring system. We create a function, $f(\gamma_{\text{out}}, \omega_{0\text{out}}, N)$, that creates a system matrix with fixed values for N and γ_{out}, and the variable, $\omega_{0\text{out}}$. It then calculates the eigenvalues of the system matrix and returns the imaginary part of the eigenvalue with the lowest frequency. We can then solve the equation $f(\gamma_{\text{out}}, \omega_{0\text{out}}, N) - \mu_1 = 0$ numerically to find the value of $\omega_{0\text{out}}$ where $f(\gamma_{\text{out}}, \omega_{0\text{out}}, N) = \mu_1$.

Once we have values for $\omega_{0\text{out}}$ and γ_{out}, we can choose any value for the mass, m, of each mass element. We then calculate the spring stiffness coefficients as $k = \omega_{0\text{out}}^2 m$ and the damping coefficients as $Z = \gamma_{\text{out}} m$.

Suppose that for our mass-spring system we want to simulate a string using 5 masses ($N = 5$), with a fundamental frequency of 440 Hz ($F = 2\pi \times 440$), a time constant of 1 second for the fundamental frequency ($\tau = 1$), and a sample rate of 6000 samples per second ($h = 1/6000$). Our algorithm produces the values

$$\begin{aligned} \omega_{0\text{out}} &= 5293.239300336853, \\ \gamma_{\text{out}} &= 7.46285773640857. \end{aligned} \tag{90}$$

We choose 1.0 for the mass making $k = \omega_{0\text{out}}^2$ and $Z = \gamma_{\text{out}}$ as the constants for the system. These values are used for each mass, spring, and damper, respectively, in the system.

Table 1 shows the eigenvalues of the system in the left hand column. The digital frequencies and damping are then calculated according to (45) and (63). The digital frequency in Hertz and the digital time constant are shown in the last two columns. We note that the digital frequency and time constant of the first eigenvalue are the values we intended. The poles of the system are plotted in Figure 21 against the regions of the s-plane. We can see that the system is stable, since all the poles are in the region where z is complex.

We built a mass-spring system, following the method outlined in Sections 2.2 and 2.3, using the coefficients calculated above and set it vibrating by displacing one of the masses. We played the resulting sound file along with a pure sine wave of 440 Hz. There was no dissonance between the pure sine wave and the frequencies produced by the simulated string. If two tones have frequencies that are similar, but not exactly the same, *beats* or fluctuations in amplitude, are heard. In our simulation no beats were heard indicating that the fundamental frequency was extremely close to 440 Hz. We then altered the system, using values for k and Z that did not correct for frequency warping and digital damping, that is, the lowest frequency eigenvalue was $-1 + 440 \times 2\pi i$. This time, when we played the sound file from the simulation, beats could be clearly heard, indicating that the lowest frequency was not 440 Hz. Using equation (45), we find the actual frequency produced by this system is 444.026 Hz, an error of 0.915%. We can verify this by playing the results of the simulation along with a sine wave of 444.026 Hz. This time no beats are heard.

In this section we have given an example of how to use the frequency warping and numerical damping equations developed in Section 3 to build a mass-spring system with prescribed behavior. We do this by building a system with eigenvalues that, after the frequency warping and numerical damping have occurred, will have the desired values. We have also shown how the frequency warping and numerical damping can be used to accurately predict the behavior of a mass-spring system implemented with the symplectic Euler method and that if these effects are not taken into consideration, the results produced by the system will not be as expected.

5. Conclusions

We have now answered the questions we asked in the introduction.

(i) By (54), symplectic Euler method is stable when

$$\omega_0 \leq \frac{1}{h}\sqrt{4 - 2\gamma h}, \tag{91}$$

where $\omega_0 = \sqrt{k/m}$, k is the spring stiffness, m is the mass, h is the time step, and γ is the damping coefficient Z divided by the mass. We can solve this equation for h to find the largest time step to ensure stability for given values of k, m, and Z. The equation is

$$h \leq \frac{Z}{m} + \sqrt{\left(\frac{Z}{m}\right)^2 + 4\frac{m}{k}}. \tag{92}$$

By (45), the symplectic Euler method warps the frequency of the analog system according to

$$\omega_d = \tan^{-1}\left(\frac{\omega_0 h\sqrt{4 - (\omega_0 h)^2 - 2\gamma h - (\gamma/\omega_0)^2}}{2 - (\omega_0 h)^2 - \gamma h}\right). \tag{93}$$

By (63), the symplectic Euler method's effect on the damping of the analog system is

$$\sigma_d = \frac{1}{h}\ln r. \tag{94}$$

input: The desired fundamental frequency F, the desired time constant τ, the number of masses N, and the length of the time step h.
output: The value used to calculate the spring stiffness coefficients $\omega_{0\text{out}}$, and the value used to calculate the damping coefficients γ_{out}.
(1) Calculate real part of the eigenvalue with lowest frequency: $\sigma_1 = -(1/2h)(1 - e^{-2h/\tau})$;
(2) Find the damping/mass coefficient for mass-spring system:
find γ_{out} such that $g(\gamma_{\text{out}}, \omega_{0\text{est}}, N) - \sigma_1 = 0$, where $\omega_{0\text{est}}$ is a rough estimate of ω_0;
(3) Calculate imaginary part of the eigenvalue with lowest frequency:
$\mu_1 = W^{-1}(F, \gamma, h)$ where $\gamma = -2\sigma_1$;
(4) Find the $\sqrt{\text{stiffness/mass}}$ value for mass-spring system:
find $\omega_{0\text{out}}$ such that $f(\gamma_{\text{out}}, \omega_{0\text{out}}, N) - \mu_1 = 0$;

Algorithm 1: Calculate system coefficients for vibrating string.

input: The digital frequency ω_d, the damping coefficient γ, the length of the time step h.
output: The analog frequency μ.
(1) $[R_1, R_2] = (-b \pm \sqrt{b^2 - 4ac})/2a$ where $a = \tan^2(\omega_d) + 1$, $b = 2\gamma h - 4 + \tan^2(\omega_d)(2\gamma h - 4)$, and $c = (\gamma h)^2 + \tan^2(\omega_d)(4 - 4\gamma h + (\gamma h)^2)$;
(2) $\mu = \begin{cases} \omega_d h \leq \dfrac{\pi}{2} & \dfrac{\sqrt{R_2^2 - \gamma^2/4}}{h} \\ \omega_d h > \dfrac{\pi}{2} & \dfrac{\sqrt{R_1^2 - \gamma^2/4}}{h}; \end{cases}$

Algorithm 2: $\mathbf{W^{-1}}(\boldsymbol{\omega_d}, \boldsymbol{\gamma}, \mathbf{h})$: calculate inverse frequency warping.

input: Damping/mass: γ, $\sqrt{\text{stiffness/mass}}$: ω_0, number of masses: N.
output: The imaginary part of the eigenvalue with the lowest frequency: imag(eig_1).
(1) Create the mass matrix (the identity matrix): $M = I_N$; //The mass matrix, M, is the N dimensional identity matrix. We are dividing through by m to normalize the mass.
(2) $k = \omega_0^2$; // If masses are normalized to 1, $\omega_0^2 = k/m = k$.
(3) $K = \begin{pmatrix} 2k & -k & 0 & \cdots & 0 \\ -k & 2k & -k & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & -k & 2k \end{pmatrix}$; // K is an $N \times N$ tridiagonal matrix with $2k$ on the diagonal, and $-k$ above and below the diagonal.
(4) $Z = \begin{pmatrix} 2\gamma & -\gamma & 0 & \cdots & 0 \\ -\gamma & 2\gamma & -\gamma & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & -\gamma & 2\gamma \end{pmatrix}$; // Z is an $N \times N$ tridiagonal matrix with 2γ on the diagonal, and $-\gamma$ above and below the diagonal.
(5) $A = \begin{pmatrix} \mathbf{0} & | & I \\ \text{------} & | & \text{------} \\ -M^{-1}K & | & -M^{-1}Z \end{pmatrix}$;
(6) eigenValues = calculateEigenValues(A);
(7) eig_1 = element of eigenValues with the smallest imaginary part;
(8) RETURN imag(eig_1); // return the imaginary part of the eigenvalue with the lowest frequency.

Algorithm 3: $\mathbf{f}(\boldsymbol{\gamma}, \boldsymbol{\omega_0}, \mathbf{N})$: find the imaginary part of eigenvalue with the lowest frequency.

This is identical to f—Algorithm 3—except that it returns the real part of the eigenvalue with the lowest frequency.

Algorithm 4: $\mathbf{g}(\boldsymbol{\gamma}, \boldsymbol{\omega_0}, \mathbf{N})$: find the real part of eigenvalue with the lowest frequency.

Table 1: Eigenvalues of the system.

Analog	Digital			
Eigenvalue	ω_d rads/sec	σ_d /sec	Freq. Hz	τ sec
$-0.99983335 + 2739.98210i$	2764.60154	−1.000000	440.00000	1.000000
$-3.73142886 + 5293.23798i$	5483.76570	−3.733751	872.76842	0.2678272
$-7.46285773 + 7485.76708i$	8089.32370	−7.472156	1287.45585	0.1338302
$-11.19428660 + 9168.15257i$	10447.41229	−11.215224	1662.75731	0.0891645
$-13.92588212 + 10225.74360i$	12263.63276	−13.958304	1951.81777	0.0716419

Here $r = |z|$ and z is calculated from (37):

$$z = \frac{1}{2}\left(2 - (\omega_0 h)^2 - \gamma h \pm \omega_0 h\sqrt{(\omega_0 h)^2 + 2\gamma h + \left(\frac{\gamma}{\omega_0}\right)^2 - 4}\right). \tag{95}$$

These results are necessary if we want to precisely understand the effect the numerical method—in this case the symplectic Euler method—has on the mathematical model of the mass-spring system. From the eigenvalues of the analog system, which represent the frequency and damping values of the mathematical model, we can determine whether the symplectic Euler implementation of the system will be stable, and if so, what the frequency and damping values of the discretized system will be. We give an example of a simulated vibrating string, created from series of masses, springs, and dampers and show how to calculate the system coefficients that result in a predetermined fundamental frequency and time constant. This example shows how knowledge of the effects of the numerical method can be used in simulating sound producing instruments.

We built a simple model of a guitar by creating six simulated strings each containing 80 masses. Using the methods presented in Section 4.3 we derived the values for Z, k, and m for each string so that the guitar had the correct tuning. The fretted notes—those created when the player presses his or her finger against the fingerboard at a certain fret—were created by stopping the mass that is closest to the calculated position of the fret from vibrating. This in effect shortens the length of the string and raises its pitch. Because there are a finite number of masses the pitch is somewhat inaccurate, but with 80 masses per string the tuning discrepancies are barely noticeable. The guitar is "played" by means of a script file that contains a series of sound events, such as plucking a particular string, placing a finger on a particular fret on a particular string, removing a finger from a fret and damping a string. Each event happens at a specified time.

The results of the simulation were surprisingly good, considering the simplicity of the model. Only the strings were simulate, not the body of the guitar. This made it sound more like a solid body electric guitar than an acoustic guitar. We could imitate the pickup position of the guitar by choosing which mass's vibration is used to create the output of the simulation. The closer the mass is to end of the string, the brighter the tone is, that is, the more high frequency components it has. By using the sum of the displacements of several masses as the sound output, a richer tone can be created.

References

[1] J. Backus, *The Acoustical Foundations of Music*, W. W. Norton, New York, NY, USA, 1969.

[2] M. Pearson, *Synthesis of organic sounds for electroacoustic music*, Ph.D. thesis, Department of Electronics, University of York, Heslington, UK, 2000.

[3] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-time modelling of musical instruments," *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, 2006.

[4] C. Cadoz, A. Luciani, and J. L. Florens, "Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the CORDIS system," *Computer Music Journal*, vol. 8, no. 3, pp. 60–73, 1984.

[5] C. Cadoz, A. Luciani, and J. L. Florens, "CORDIS-ANIMA. A modeling and simulation system for sound and image synthesis. The general formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, 1993.

[6] C. Cadoz and J. L. Florens, "The physical model: modeling and simulating the instrumental universe," in *Representations of Musical Signals*, G. De Poli, A. Piccialli, and C. Roads, Eds., MIT Press, Cambridge, Mass, USA, 1991.

[7] F. Avanzini, *Computational issues in physically-based sound models*, Ph.D. thesis, Department of Computer Science and Electronics, University of Padova, Padova, Italy, 2001.

[8] G. De Poli and D. Rocchesso, "Physically based sound modelling," *Organised Sound*, vol. 3, no. 1, pp. 61–76, 1998.

[9] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, Springer, New York, NY, USA, 2002.

[10] M. Beck, *Symplectic methods applied to the Lotka-Volterra system*, M.S. thesis, McGill University, Montreal, Canada, 2003.

[11] D. Morgan and S. Qiao, "Accuracy and stability in mass-spring systems for sound synthesis," in *Proceeding of the Canadian Conference on Computer Science & Software Engineering (C3S2E '08)*, pp. 69–80, Montreal, Canada, May 2008.

[12] B. J. Leimkuhler, S. Reich, and R. D. Skeel, "Integration methods for molecular dynamics," in *Mathematical Approaches to Biomolecular Structure and Dynamics*, vol. 82 of *IMA Volumes in Mathematics and Its Applications*, pp. 161–185, Springer, New York, NY, USA, 1996.

[13] M. J. Duncan, "The longterm dynamical evolution of orbital configurations," in *Computational Astrophysics; 12th Kingston Meeting on Theoretical Astrophysics*, D. A. Clarke and M. J. West, Eds., vol. 123 of *Astronomical Society of the Pacific Conference Series*, pp. 17–19, Astronomical Society of the Pacific, San Francisco, Calif, USA, 1997.

[14] H. Kinoshita, H. Yoshida, and H. Nakai, "Symplectic integrators and their application to dynamical astronomy," *Celestial Mechanics and Dynamical Astronomy*, vol. 50, no. 1, pp. 59–71, 1990.
[15] W. E. Boyce and R. C. DiPrima, *Elementary Differential Equations and Boundary Value Problems*, John Wiley & Sons, New York, NY, USA, 2001.
[16] M. Pearson, "Tao: a physical modelling system and related issues," *Organised Sound*, vol. 1, no. 1, pp. 43–50, 1996.
[17] J. L. Florens, "Real time Bowed String Synthesis with Force Feedback Gesture Interaction," Invited paper, Forum Acousticum, 2002.
[18] E. A. Lee and P. Varaiya, *Structure and Interpretation of Signals and Systems*, Addison Wesley, Reading, Mass, USA, 2003.
[19] R. C. Dorf, *The Engineering Handbook*, CRC Press, Boca Raton, Fla, USA, 2004.
[20] K. Steiglitz, *A Digital Signal Processing Primer: With Applications to Digital Audio and Computer Music*, Addison Wesley, Reading, Mass, USA, 1996.
[21] L. Meirovitch, *Fundamentals of Vibrations*, McGraw Hill, New York, NY, USA, 2001.
[22] J. Dwight Aplevich, *The Essentials of Linear State Space Systems*, John Wiley & Sons, New York, NY, USA, 2000.
[23] M. R. Hatch, *Vibration Simulation Using MATLAB and ANSYS*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2001.
[24] A. P. French, *Vibrations and Waves*, W. W. Norton, New York, NY, USA, 1971.